

Remarks

Applicant acknowledges with thanks Examiner Jyoti Dave's assistance in granting a telephone interview with Thomas Plunkett on September 17, 2008, during the course of which interview the participants generally discussed the claimed embodiments. The participants appeared to reach agreement that the combination of references does not disclose the features shown in Figure 4, and defined by the currently amended claims. Examiner Dave indicated that further consideration of the application would follow upon receipt of this Reply.

I. Summary of Rejections

Prior to the Office Action mailed June 25, 2008, Claims 1-10 were pending in the application. In the Office Action, Claims 1-10 were rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over Claims 1-10 of U.S. Patent Application 2004/0215594. Claims 6-10 were rejected under 35 U.S.C. 112 for insufficient antecedent basis. Claims 1-10 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Doolittle et al. (U.S. Patent No. 6,898,617, hereinafter Doolittle) in view of Parallel Execution: Oracle8i Concepts Release 8.1.5 (1999, hereinafter Oracle8i) and in further view of Motomura (U.S. Patent No. 5,815,727, hereinafter Motomura).

II. Summary of Applicant's Amendment

The present Reply amends Claims 1, 2, 4-8, and 10; cancels Claim 9; and adds new Claims 11-21, all as shown above. Reconsideration of the Application is respectfully requested.

III. Rejections under Non-statutory Obviousness-type Double Patenting

In the Office Action, Claims 1-10 were rejected on the ground of non-statutory obviousness-type double patenting as being unpatentable over Claims 1-10 of U.S. Patent Application 2004/0215594.

Accordingly, provided herewith is a terminal disclaimer compliant with 1.321(c). Applicant respectfully submits that the filing of a terminal disclaimer renders moot the rejection of the claims under non-statutory obviousness-type double patenting, and reconsideration is respectfully requested.

IV. Rejections Under 35 U.S.C. 112

In the Office Action, Claims 6, 7, and 10 were rejected under 35 U.S.C. 112 for insufficient antecedent basis. Claims 8 and 9 were rejected as depending upon a rejected base Claim. Accordingly, Claims 6, 7, and 10 have been amended to correct these issues.

Applicant respectfully submits that Claims 6-10 as amended satisfy the requirements of 35 U.S.C. 112, and reconsideration thereof is respectfully requested.

V. Rejections Under 35 U.S.C. 103(a)

Claim 1

Claim 1 (as amended) defines:

1. A method for implementing a two-phase commit protocol, comprising:
dispatching a first prepare operation from a first server thread to a second server thread, the first prepare operation associated with a first resource and a prepare phase;
processing a second prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, the second prepare operation associated with a second resource and the prepare phase;
determining that the prepare phase is complete;
dispatching a first commit operation from the first server thread to a third server thread, the first commit operation associated with the first resource and a commit phase;
processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread, the second commit operation associated with the second resource and the commit phase; and
after determining that the commit phase is complete, writing results of the commit phase to a transaction log.

Claim 1 has been amended to more clearly define the embodiment described therein.

In the Office Action, Claim 1 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Doolittle in view of Oracle8i, and further in view of Motomura.

Doolittle discloses how deadlock situations within a computing environment are avoided by properly managing pools of threads used to service requests of the computing environment. When a server of the computing environment receives a request to be processed and that request is waiting on a response from a client of the computing environment, the set of eligible thread pools for the response is dynamically altered. This dynamic altering allows the response to be serviced by a thread pool different from the thread pool servicing the request, thereby avoiding a deadlock situation. (Abstract).

Doolittle discloses that when a client sends a request to a server to be processed, the server obtains an available thread from a selected thread pool in order to process the request. Doolittle further discloses that before a client communicates with a server, a client session is created, as described with reference to FIG. 4a. (Col. 4, lines 44-60).

Oracle 8i discloses that to execute a DML operation in parallel, the parallel execution coordinator acquires or spawns parallel execution servers and each parallel execution server executes a portion of the work under its own parallel process transaction. Each parallel execution server creates a different parallel process transaction. (Page 36, lines 1-3).

Motomura discloses executing a program consisting of a plurality of threads in parallel, including a thread generating portion for managing three thread states of executing state, executable state and waiting state. A thread execution control portion controls execution of the threads so as to perform all of the synchronization operations in the program being performed only in the direction from the thread having the smaller virtual thread number to the thread having the greater virtual thread number. (Abstract).

Claim 1, as amended, defines dispatching a first prepare operation from a first server thread to a second server thread, the first prepare operation associated with a first resource and a prepare phase; processing a second prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, the second prepare operation associated with a second resource and the prepare phase; and determining that the prepare phase is complete. Claim 1 has been further amended to define dispatching a first commit operation from the first server thread to a third server thread, the first commit operation associated with the first resource and a commit phase; processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread, the second commit operation associated with the second resource and the commit phase; and after determining that the commit phase is complete, writing results of the commit phase to a transaction log.

Applicant respectfully submits that Doolittle teaches dispatching one or more requests to one or more service threads, and queuing up any requests for which sufficient threads are not available. In Doolittle, the thread that dispatches requests to one or more service threads does not itself perform any processing other than the dispatching. Since Doolittle's dispatching thread does not itself perform any processing other than the dispatching, Doolittle's dispatching thread does not disclose Applicant's first server thread which dispatches a first prepare operation and processes a second prepare operation in parallel while the first prepare operation is being processed. Applicant respectfully submits that Doolittle does not disclose or render obvious dispatching a first prepare operation from a first server thread to a second server thread, processing a second prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, dispatching a first commit operation from the first server thread to a third server thread, and processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread.

Applicant respectfully submits that Oracle 8i discloses that each parallel execution server creates a different parallel process transaction. (Page 36). Applicant respectfully submits that the Oracle 8i discloses different parallel process transactions being performed in parallel on different parallel execution servers.

Motomura discloses executing a program consisting of a plurality of threads in parallel. (Abstract).

In the Office Action, it was asserted that it would have been obvious to utilize the teachings of references such as Motomura to provide for parallel processing by a plurality of threads in parallel. It was

further asserted that it would have been obvious to one of skill in the art to use multiple threads to execute parallel processing of two phase commit, such as disclosed by Oracle8i. It was further asserted that it would have been obvious to one skilled in the art to use the methods disclosed in Doolittle to execute the parallel processing using multiple threads.

As discussed above, Applicant respectfully submits that the method disclosed in Doolittle is different from the steps of Claim 1 as amended. Furthermore, Applicant respectfully submits that Oracle 8i discloses performing parallel process transactions in parallel on different parallel execution servers. However, Oracle 8i does not appear to disclose performing prepare and commit phase operations in parallel. Furthermore, Applicant respectfully submits that Motomura is silent regarding performing prepare and commit phase operations in parallel. As such, Applicant respectfully submits that the combination of references does not disclose a first server thread processing the second prepare operation and the second commit operation, a second server thread processing the first prepare operation, and a third server thread processing the first commit operation. Applicant respectfully submits that the combination of references does not disclose dispatching a first prepare operation from a first server thread to a second server thread, processing a second prepare operation by the first server thread in parallel to the first prepare operation being processed by the second server thread, dispatching a first commit operation from the first server thread to a third server thread, and processing a second commit operation by the first server thread in parallel to the first commit operation being processed by the third server thread.

In view of the comments provided above, Applicant respectfully submits that the embodiment defined by Claim 1, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claim 6

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Claim 6 has been similarly amended by the current Reply to more clearly define the embodiments therein. For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 6, as amended, is likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 2-5 and 7-10

Claim 9 has been canceled, rendering moot the rejection of this claim. Claims 2-5, 7-8, and 10 depend from and include all of the features of Claims 1 or 6. Claims 2-5, 7-8, and 10 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim,

Application No: 10/762,944
Office Action mailed: June 25, 2008
Reply to Office Action dated: October 27, 2008

and further in view of the amendments to the independent claims, and the comments provided above. Reconsideration thereof is respectfully requested.

VI. Additional Amendments

Claims 11-21 have been newly added by the present Reply. Applicant respectfully requests that new Claims 11-21 be included in the application and considered therewith.


VII. Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. § 1.136 for extending the time to respond up to and including today, October 27, 2008.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

By: 
Thomas K. Plunkett
Reg. No. 57,253

Date: October 27, 2008

Customer No. 80548
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800